

Optimizing MDS Coded Caching in Wireless Networks with Device-to-Device Communication

Jesper Pedersen[†], Alexandre Graell i Amat[†], Iryna Andriyanova[‡], and Fredrik Brännström[†]

[†]Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden

[‡]ETIS Laboratory, ENSEA/University of Cergy-Pontoise/CNRS, Cergy-Pontoise, France

Abstract—We consider the caching of content in the mobile devices in a wireless network using maximum distance separable (MDS) codes. We focus on a small cell, served by a base station (BS), where mobile devices arrive and depart according to a Poisson birth-death process. Users requesting a particular file download coded symbols from caching devices using device-to-device communication. If additional symbols are required to decode the file, these are downloaded from the BS. We optimize the MDS codes to minimize the network load under a global average cache size constraint. We show that, for most practical scenarios, using optimized MDS codes results in significantly lower network load compared to when caching the most popular files. Furthermore, our results show that caching coded symbols of each file on all mobile devices, i.e., maximal spreading, is optimal.

I. INTRODUCTION

Mobile data traffic is predicted to grow by 800% in the next 6 years [1]. This imposes a severe strain on existing wireless networks. One of the most promising methods to reduce downlink traffic is to store content closer to end users, commonly referred to as *caching* [2]. Caching has been studied extensively in recent years. In [3], a central server with access to a file library serves a number of users, each with the ability to cache content, using a common channel. It is shown that, exploiting local caches as side information, transmitting coded multicasts, i.e., linear combinations of packets from different files, can significantly reduce the amount of data that is necessary to transmit over the channel. This concept is investigated in [4] for the specific application of wireless networks. The central server is omitted and requests for content are served by mobile devices caching data, which broadcast interfile coded messages using device-to-device (D2D) communication.

In wireless networks, mobile devices that cache content might move out of range. The impact of data loss when mobile devices leave the network is considered in [5]–[7]. Instantaneous repair of single device departures is investigated in [5] and scheduled repair of lost content due to multiple devices leaving the network is treated in [6] and [7]. The papers [5]–[7] consider a single file and it is shown that the use of erasure correcting codes help to reduce the amount of data that is downloaded from the base station (BS). In [8], multiple files are cached in a number of small BSs from which mobile devices download content. It is shown that caching content

using maximum distance separable (MDS) codes reduces the macro BS downlink traffic.

In this paper, we consider the use of MDS codes to cache content in the mobile devices of a wireless cellular network. The devices roam between neighboring cells according to a Poisson random process, similar to the modeling of device mobility in [5]–[7], and request files from a library at random times. Files are encoded using MDS codes of equal code length but potentially different code rate, and coded symbols are cached in a number of mobile devices. We formulate a mixed integer linear program (MILP) to find the MDS codes that minimize the network load, assuming a global average cache size constraint (across all devices). We show that optimizing the rates of the MDS codes may yield a significantly lower network load than when caching only the most popular files. We also show that, for the considered scenario, caching symbols of a given file on all mobile devices, i.e., maximal spreading [9], is optimal.

Notation: $\mathbb{P}(\cdot)$ and $\mathbb{E}[\cdot]$ denote probability and expectation, respectively. A random variable X that follows the binomial distribution with parameters n and p is denoted by $X \sim \mathcal{B}(n, p)$. We use bold lowercase letters \mathbf{x} to denote vectors. $\mathbf{0}_m$ and $\mathbf{1}_m$ are the length- m vectors with all entries 0 and 1, respectively.

II. SYSTEM MODEL

We consider a large area of A_{tot} square units (sq.u.) consisting of many small cells with equal area A sq.u., each served by a BS. M_{tot} mobile devices are uniformly spread over the large area, which gives that the number of devices in a given small cell, denoted by X , is binomially distributed as

$$X \sim \mathcal{B}(M_{\text{tot}}, A/A_{\text{tot}}). \quad (1)$$

The expected number of devices initially available in a small cell, denoted by M , is

$$M = \mathbb{E}[X] = M_{\text{tot}} \cdot A/A_{\text{tot}}. \quad (2)$$

Users wish to download files from a library of N files that is always available at the BSs. We assume that all files have equal size and that the file popularity follows the Zipf distribution [10], i.e., the popularity of file i is given by

$$p_i = \frac{1/i^\sigma}{\sum_{\ell=1}^N 1/\ell^\sigma}, \quad i = 1, \dots, N, \quad \sigma \geq 0, \quad (3)$$

where σ is the skewness parameter of the distribution.

Content allocation: Each file i that is to be cached is partitioned into k_i packets and encoded into n symbols using an (n, k_i) MDS code [11]. The n coded symbols are cached in n mobile devices (possibly different for each file) in the area, chosen uniformly at random. Hence, for each file i , the n devices caching the file store a fraction $\alpha_i = 1/k_i$ of the file. Thus,

$$\alpha_i \in \{0\} \cup \left\{ \frac{1}{k_i} : k_i = 1, \dots, n \right\} \triangleq \mathcal{A}, \quad i = 1, \dots, N, \quad (4)$$

where $\alpha_i = 0$ implies that file i is not cached. We define the vector $\alpha = (\alpha_1, \dots, \alpha_N)$ and refer to it as the *content allocation*.

The number of caching devices in a given small cell, denoted by Y , is binomially distributed as

$$Y \sim \mathcal{B}(n, A/A_{\text{tot}}), \quad (5)$$

and the expected number of caching devices in a given small cell, denoted by m , is

$$m = \mathbb{E}[Y] = n \cdot A/A_{\text{tot}}. \quad (6)$$

Arrival-departure process: Mobile devices roam in and out of a small cell. We assume that the arrival of mobile devices caching a coded symbol of a given file and the arrival of devices not caching a symbol of the file are two independent arrival processes with interarrival times exponentially independent, identically distributed (i.i.d.) with rates λ_1 and λ_2 , respectively, per time unit (t.u.⁻¹). Therefore, the overall arrival rate is $\lambda = \lambda_1 + \lambda_2$. Each device has an exponential i.i.d. lifetime with rate μ t.u.⁻¹ before departing the small cell. Hence, the number of devices in the cell follows a Poisson birth-death process with rate λ/μ [12]. We assume $\lambda_1 = m\mu$ and $\lambda_2 = (M - m)\mu$ and, hence, the expected number of devices and the expected number of caching devices in the small cell is M and m , respectively. This process can be characterized by an M/M/ ∞ queueing model where the instantaneous probability of having j caching devices in the cell is [12]

$$\pi_j = \frac{(\lambda_1/\mu)^j}{j!} e^{-\lambda_1/\mu} = \frac{m^j}{j!} e^{-m}, \quad j \geq 0. \quad (7)$$

Note that this model predicts a nonzero probability that the number of mobile devices in a small cell is larger than M_{tot} , which violates our initial assumption on the total number of devices in the area. However, we consider scenarios where $M \ll M_{\text{tot}}$, such that this probability is negligibly small.

Data download: Mobile devices request files at random times with the time between requests exponentially i.i.d. with rate ω t.u.⁻¹. Hence, the expected total request rate in the small cell is $M\omega$ t.u.⁻¹. A device request file i with probability p_i , given by (3). Due to the MDS property, k_i coded symbols are sufficient to decode the file [11]. The user requesting content downloads as many coded symbols as possible (up to k_i) from available caching devices and, if additional symbols are required, these are retrieved from the BS. The equivalent number of files that are downloaded from the BS per t.u. is

defined as the *downlink rate* and the equivalent number of files downloaded from caching devices (per t.u.) is defined as the *D2D communication rate*. We assume that the communication is error free and incurs zero delay.

III. NETWORK STATISTICS ANALYSIS

In this section, we derive the probability to have a number of caching devices available at the time of a request. For later use, we define \mathcal{J}_j as the event that there are j caching devices available at the time of a request for file i . We also define \mathcal{C} and $\bar{\mathcal{C}}$ as the events that the request comes from a caching device and comes from a noncaching device, respectively. We have the following proposition.

Proposition 1. *The probability of having j mobile devices caching a coded symbol of file i at the time of a request for file i is*

$$q_j = \frac{M - m + j}{M} \cdot \frac{m^j}{j!} e^{-m}, \quad j \geq 0. \quad (8)$$

Proof: The probability q_j is given by

$$q_j = \mathbb{P}(\mathcal{J}_j) = \mathbb{P}(\mathcal{J}_j|\bar{\mathcal{C}})\mathbb{P}(\bar{\mathcal{C}}) + \mathbb{P}(\mathcal{J}_j|\mathcal{C})\mathbb{P}(\mathcal{C}). \quad (9)$$

Since the number of caching devices and the number of noncaching devices are Poisson distributed with rates m and $M - m$, respectively, the expected fraction of mobile devices caching a coded symbol of file i is m/M . A request is equally likely to come from any mobile device and we have that

$$\mathbb{P}(\bar{\mathcal{C}}) = \frac{M - m}{M}, \quad (10)$$

and

$$\mathbb{P}(\mathcal{C}) = \frac{m}{M}. \quad (11)$$

Using (7),

$$\mathbb{P}(\mathcal{J}_j|\bar{\mathcal{C}}) = \pi_j. \quad (12)$$

If there is a request for file i from a caching device (event \mathcal{C}) there is at least one caching device in the cell. The probability that there are j caching devices in the cell is the probability that $j - 1$ additional caching devices are available, i.e.,

$$\mathbb{P}(\mathcal{J}_j|\mathcal{C}) = \frac{m^{j-1}}{(j-1)!} e^{-m} = \frac{j\pi_j}{m}, \quad (13)$$

using (7). Using (10)–(13) with (7) in (9), we get (8). ■

The amount of data that is downloaded from the BS and from the mobile devices is given by the following proposition.

Proposition 2. *Provided that there are j mobile devices caching a coded symbol of file i in the small cell, then, for a given request for file i , the fraction of the file that is downloaded from the BS is*

$$\gamma_{ij}^{(\text{BS})} = \begin{cases} 1 - j\alpha_i, & \text{if } 0 \leq j < 1/\alpha_i \\ 0, & \text{if } j \geq 1/\alpha_i \end{cases}. \quad (14)$$

$$h(\alpha) = M\omega \sum_{i=1}^N \sum_{j=0}^{\infty} p_i \max \left\{ j\alpha_i \left((1-2\theta)q_j - (1-\theta)\frac{\pi_j}{M} \right) + \theta q_j, (1-\theta) \left(q_j - \alpha_i \frac{j\pi_j}{M} \right) \right\} \quad (23)$$

The fraction of file i that is downloaded from caching devices if the request comes from a mobile device not caching a coded symbol of file i is

$$\gamma_{ij}^{(\text{D2D}, \bar{C})} = \begin{cases} j\alpha_i, & \text{if } 0 \leq j < 1/\alpha_i \\ 1, & \text{if } j \geq 1/\alpha_i \end{cases} \quad (15)$$

and if the request comes from a device caching a symbol of file i

$$\gamma_{ij}^{(\text{D2D}, C)} = \begin{cases} (j-1)\alpha_i, & \text{if } 1 \leq j < 1/\alpha_i \\ 1 - \alpha_i, & \text{if } j \geq 1/\alpha_i \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

Proof: We require k_i coded symbols to decode file i . If at the time of a request for file i there are $j < k_i = 1/\alpha_i$ devices caching a symbol of file i in the small cell, $k_i - j$ symbols are retrieved from the BS. If $j \geq k_i$, no symbols are downloaded from the BS. Hence, the fraction of file i that is downloaded from the BS is given by (14). Following the same argument, it is trivial to obtain (15) and (16). ■

The expected downlink rate for a content allocation α , denoted by $f(\alpha)$, is given by

$$f(\alpha) = M\omega \sum_{i=1}^N \sum_{j=0}^{\infty} p_i \gamma_{ij}^{(\text{BS})} q_j, \quad (17)$$

since the amount of data that is downloaded from the BS is the same, irrespective of whether the request comes from a caching or a noncaching device. The expected D2D communication rate, denoted by $g(\alpha)$, is given by

$$g(\alpha) = M\omega \sum_{i=1}^N \sum_{j=0}^{\infty} p_i \left(\gamma_{ij}^{(\text{D2D}, \bar{C})} \mathbb{P}(\mathcal{J}_j, \bar{C}) + \gamma_{ij}^{(\text{D2D}, C)} \mathbb{P}(\mathcal{J}_j, C) \right), \quad (18)$$

where, using (10)–(13), (15), and (16),

$$\begin{aligned} & \gamma_{ij}^{(\text{D2D}, \bar{C})} \mathbb{P}(\mathcal{J}_j, \bar{C}) + \gamma_{ij}^{(\text{D2D}, C)} \mathbb{P}(\mathcal{J}_j, C) \\ &= \begin{cases} j\alpha_i \mathbb{P}(\mathcal{J}_j, \bar{C}) + (j-1)\alpha_i \mathbb{P}(\mathcal{J}_j, C), & \text{if } 0 \leq j < 1/\alpha_i \\ \mathbb{P}(\mathcal{J}_j, \bar{C}) + (1 - \alpha_i) \mathbb{P}(\mathcal{J}_j, C), & \text{if } j \geq 1/\alpha_i \end{cases} \\ &= \begin{cases} j\alpha_i (q_j - \frac{\pi_j}{M}), & \text{if } 0 \leq j < 1/\alpha_i \\ q_j - \alpha_i \frac{j\pi_j}{M}, & \text{if } j \geq 1/\alpha_i \end{cases}. \end{aligned} \quad (19)$$

IV. MINIMIZING THE WEIGHTED COMMUNICATION RATE

We consider the optimization of the content allocation α such that the weighted communication rate

$$h(\alpha) \triangleq \theta f(\alpha) + (1-\theta)g(\alpha) \quad (20)$$

is minimized. Minimizing the expected downlink rate corresponds to $\theta = 1$. However, it might be desirable to also limit the D2D communication rate for various reasons, such as device battery constraints and interference between devices.

Therefore, we consider $0.5 \leq \theta \leq 1$, where $\theta \geq 0.5$ stems from the fact that the bottleneck is in the downlink.

We enforce a global average cache size constraint, denoted by β ,¹ where

$$\sum_{i=1}^N \alpha_i \leq \beta. \quad (21)$$

This implies an average cache size constraint per device $\bar{\beta}_d = \beta n / M_{\text{tot}}$. The optimization problem can be formulated as

$$\underset{\alpha_i \in \mathcal{A}}{\text{minimize}} \quad h(\alpha) \quad (22a)$$

$$\text{subject to} \quad \sum_{i=1}^N \alpha_i \leq \beta \quad (22b)$$

We denote by α^* the *optimal* content allocation resulting from (22).

The objective function (22a) can be rewritten as shown in (23) (at the top of the page) using (14) and (17)–(20). It is clear that the objective function (23) is a sum of piecewise linear functions of α_i . This allows us to rewrite the optimization problem in a way that is tractable, using the epigraph formulation [13]. Using (23) and introducing a new optimization variable $t_{ij} \in \mathbb{R}$, the optimization problem (22) can be written equivalently as

$$\underset{\substack{\alpha_i \in \mathcal{A} \\ t_{ij} \in \mathbb{R}}}{\text{minimize}} \quad \sum_{i=1}^N \sum_{j=0}^{\infty} t_{ij} \quad (24a)$$

$$\text{subject to} \quad \sum_{i=1}^N \alpha_i \leq \beta \quad (24b)$$

$$t_{ij} + p_i j \left((2\theta - 1)q_j + (1-\theta)\frac{\pi_j}{M} \right) \alpha_i \geq \theta p_i q_j \quad (24c)$$

$$t_{ij} + (1-\theta)p_i \frac{j\pi_j}{M} \alpha_i \geq (1-\theta)p_i q_j \quad (24d)$$

where the constraints (24c) and (24d) arise from the first and second term in the max function in (23). Note that we drop the $M\omega$ term in (24a) since it is irrelevant to the solution of the optimization problem.

The α_i can only take on the discrete values given by (4). To handle this, we introduce the binary optimization variable $b_{i\ell} \in \{0, 1\}$ and let

$$\alpha_i = b_{i0} \cdot 0 + b_{i1} \frac{1}{n} + b_{i2} \frac{1}{n-1} + \dots + b_{in}, \quad \forall i, \quad (25)$$

such that

$$\sum_{\ell=0}^n b_{i\ell} = 1, \quad \forall i. \quad (26)$$

¹In practice, a strict cache size constraint per device would be desirable. Unfortunately, this leads to a complicated optimization problem. Therefore, to simplify the analysis, we enforce a global average cache size constraint instead.

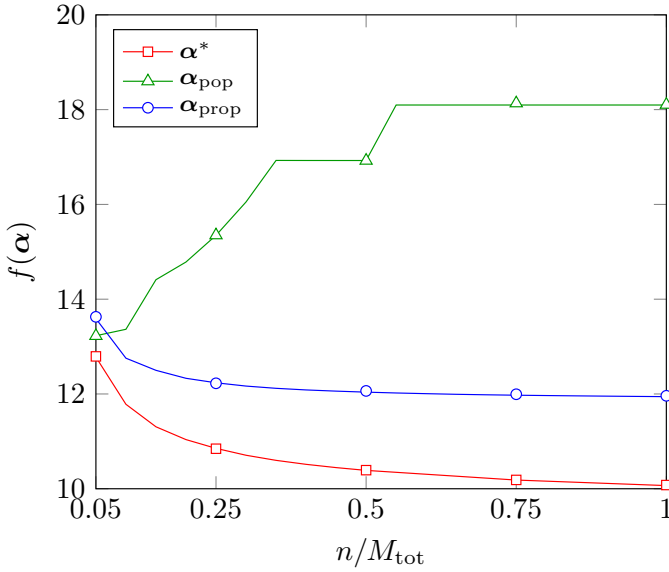


Figure 1. The downlink rate versus n/M_{tot} using different content allocations for $M = 20$ and $\theta = 1$. All markers correspond to simulated downlink rate.

Finally, we formulate an MILP that is equivalent to (22) and (24), minimizing the objective function (24a) under the constraints (24b)–(24d), (25), and (26),

$$\begin{aligned}
 & \underset{\substack{\alpha_i, t_{ij} \in \mathbb{R} \\ b_{i\ell} \in \{0,1\}}}{\text{minimize}} && \sum_{i=1}^N \sum_{j=0}^{\infty} t_{ij} \\
 & \text{subject to} && \sum_{i=1}^N \alpha_i \leq \beta \\
 & && t_{ij} + p_{ij} \left((2\theta - 1)q_j + (1 - \theta) \frac{\pi_j}{M} \right) \alpha_i \geq \theta p_i q_j \\
 & && t_{ij} + (1 - \theta) p_i \frac{j \pi_j}{M} \alpha_i \geq (1 - \theta) p_i q_j \\
 & && \alpha_i - \sum_{\ell=1}^n \frac{b_{i\ell}}{n - \ell + 1} = 0, \quad \forall i \\
 & && \sum_{\ell=0}^n b_{i\ell} = 1, \quad \forall i
 \end{aligned} \tag{27}$$

V. NUMERICAL RESULTS

In Figs. 1–3 we evaluate the downlink rate (17) and the weighed communication rate (20) for several content allocations α . Besides the optimal content allocation α^* , we consider a *popular* content allocation, where each of the $\lfloor \beta \rfloor$ most popular files is cached in n (possibly different) mobile devices (i.e., using an $(n, 1)$ repetition code). The popular content allocation is given by

$$\alpha_{\text{pop}} = (\mathbf{1}_{\lfloor \beta \rfloor}, \mathbf{0}_{N - \lfloor \beta \rfloor}). \tag{28}$$

We also consider a *proportional* content allocation,

$$\alpha_{\text{prop}} = \beta \mathbf{p}, \tag{29}$$

where $\mathbf{p} = (p_1, \dots, p_N)$. We remark that the proportional content allocation typically leads to noninteger k_i file partitions. Albeit impractical, this content allocation is interesting for

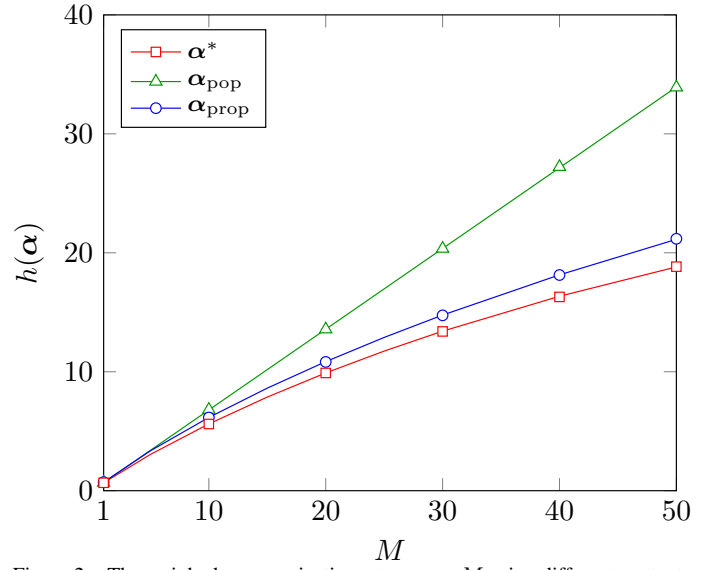


Figure 2. The weighed communication rate versus M using different content allocations for $n/M_{\text{tot}} = 1$ and $\theta = 0.75$. The markers correspond to simulated data.

comparison purposes. For the results, we consider a file library with $N = 100$ files and Zipf parameter $\sigma = 0.7$. We assume departure rate $\mu = 1$ t.u.⁻¹ and request rate $\omega = 1$ t.u.⁻¹. Note that the optimal content allocation does not depend on ω . We also assume that the average cache size constraint per device is $\bar{\beta}_d = 1$ file. Monte Carlo simulation results are also included in the figures.

Fig. 1 shows the downlink rate in (17) versus the code length n , normalized to the total number of mobile devices in the area M_{tot} , for $M = 20$. Note that $n/M_{\text{tot}} = 1$ corresponds to maximal spreading [9], i.e., storing a coded symbol of a given file on as many devices as possible. In this case, the global average cache size constraint (21) becomes a strict cache size constraint, i.e., $\beta = \bar{\beta}_d$. We recall that $n/M_{\text{tot}} = m/M$, using (2) and (6). We observe that the downlink rate decreases with n/M_{tot} for both the optimal and proportional content allocations, i.e., maximal spreading [9] is optimal. This optimality is observed for the weighed communication rate in (20) for several values of N , σ , M , $\bar{\beta}_d$, and θ . For the popular content allocation, the downlink rate increases with n/M_{tot} . This is because β decreases as n/M_{tot} increases and a smaller number of complete files can be cached in the mobile devices. For $n/M_{\text{tot}} = 1$, the downlink rate using the popular content allocation is almost 80% higher compared to when using the optimal content allocation.

In Fig. 2, we show the weighed communication rate $h(\alpha)$ in (20) versus the expected number of mobile devices in the small cell, M , for $n/M_{\text{tot}} = 1$, i.e., maximal spreading, and $\theta = 0.75$. The expected accumulated cache capacity of the devices in the small cell, $M\bar{\beta}_d$, grows with M but so does the expected request rate $M\omega$. An increase of $M\bar{\beta}_d$ leads to a decrease of $h(\alpha)$, while an increase of $M\omega$ leads to an increase of $h(\alpha)$. It is observed in the figure that $h(\alpha)$ increases with M , i.e., the effect of an increase in weighed communication rate due to the increased request rate domi-

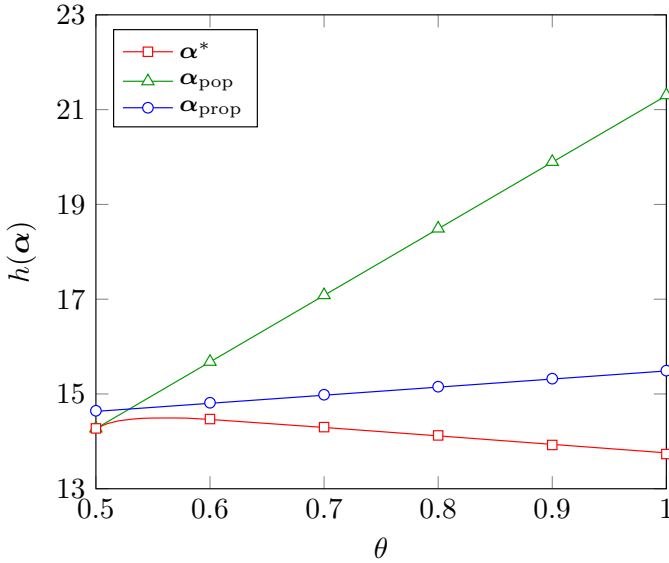


Figure 3. The weighted communication rate using the various content allocations as a function of θ for $n/M_{\text{tot}} = 1/6$ and $M = 30$. The markers correspond to simulated data.

nates the performance. For small M , $h(\alpha)$ is similar for all considered content allocations. However, for larger M , MDS coded caching with optimized code rates yields significantly lower weighted communication rate. Compared to the optimal content allocation, the popular content allocation and the proportional content allocation involve roughly a 50% and a 10% increase, respectively, in the weighted communication rate for $M = 30$.

In Fig. 3, the weighted communication rate in (20) is plotted for the considered content allocations as a function of θ for $n/M_{\text{tot}} = 1/6$ and $M = 30$. We see that the weighted communication rate increases linearly with θ for the popular and proportional content allocations. This is expected since these content allocations are obviously not affected by changing θ . The weighted communication rate for the optimal content allocation is on the other hand not monotonically increasing with θ . For $\theta = 0.5$, the weight of the downlink rate in (17) and the D2D communication rate in (18) is the same, i.e., it is equally “expensive” to download files from the BS and the caching devices. It is easy to prove that, for integer β , the optimal content allocation for $\theta = 0.5$ is the popular content allocation. However, this is not the case in general. For the example in Fig. 3, $\beta = 6$, and thus for $\theta = 0.5$ popular allocation is optimal. For large θ , the popular content allocation leads to a significant communication overhead compared to the optimal content allocation. On the other hand, the proportional content allocation incurs a small overhead compared to the optimal content allocation. Finally, in Fig. 4 we show the content allocations as a function of file index used to achieve the weighted communication rate in Fig. 3 for $\theta = 0.6$.

VI. CONCLUSIONS

We optimized the caching of content in mobile devices using maximum distance separable codes. We formulated a mixed

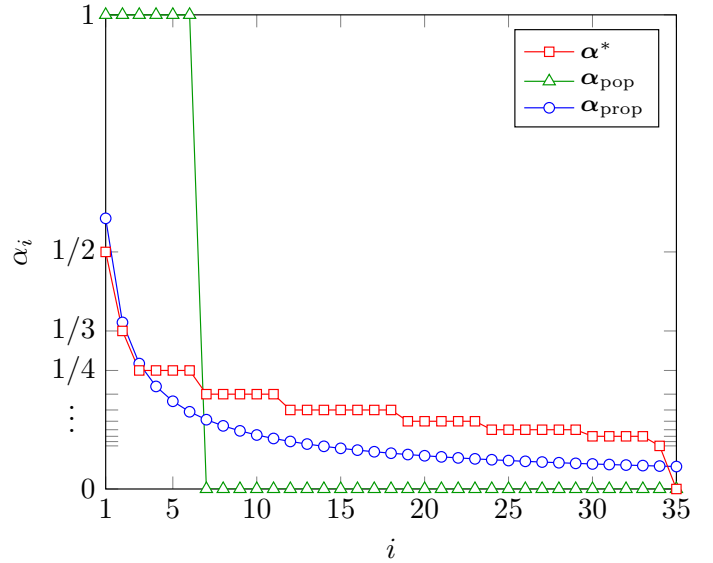


Figure 4. Various content allocations as a function of file rank for the 35 most popular files for $n/M_{\text{tot}} = 1/6$, $M = 30$, and $\theta = 0.6$.

integer linear program to minimize the weighted sum of the downlink rate and the device-to-device communication rate under a global average cache size constraint. We showed that optimized MDS coded caching may yield a significantly lower weighted communication rate compared to when caching (uncoded) the most popular files. Furthermore, our results show that caching coded symbols of a particular file on all devices, i.e., maximal spreading, is optimal.

REFERENCES

- [1] “Ericsson mobility report,” White Paper, Ericsson, Nov. 2016.
- [2] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, “Five disruptive technology directions for 5G,” *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.
- [3] M. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.
- [4] M. Ji, G. Caire, and A. Molisch, “Fundamental limits of caching in wireless D2D networks,” *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2016.
- [5] J. Pääkkönen, C. Hollanti, and O. Tirkkonen, “Device-to-device data storage for mobile cellular systems,” in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Atlanta, GA, 2013.
- [6] J. Pedersen, A. Graell i Amat, I. Andriyanova, and F. Brännström, “Repair scheduling in wireless distributed storage with D2D communication,” in *Proc. IEEE Inf. Theory Workshop (ITW)*, Jeju, Korea, 2015, pp. 69–73.
- [7] —, “Distributed storage in mobile wireless networks with device-to-device communication,” *IEEE Trans. Commun.*, vol. 64, no. 11, pp. 4862–4878, Nov. 2016.
- [8] V. Bioglio, F. Gabry, and I. Land, “Optimizing MDS codes for caching at the edge,” in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, San Diego, CA, 2015.
- [9] D. Leong, A. G. Dimakis, and T. Ho, “Distributed storage allocations,” *IEEE Trans. Inf. Theory*, vol. 58, no. 7, pp. 4733–4752, Jul. 2012.
- [10] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “Web caching and zipf-like distributions: evidence and implications,” in *Proc. IEEE 18th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, New York, NY, 1999, pp. 126–134.
- [11] W. E. Ryan and S. Lin, *Channel codes: Classical and modern*. Cambridge University Press, 2009.
- [12] S. L. Miller and D. Childers, *Probability and random processes*. Elsevier, 2004.
- [13] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2009.